

for each file in the set of files identifying a handler routine and sending each file to the identified handler routine; and

for each file in the set of files, in the identified handler routine, determining the application functionality required to execute each file.

2. (Unchanged) The method of Claim 1, wherein the application functionality comprises products, features and components.

3. (Unchanged) The method of Claim 1, further comprising the steps of:
identifying the set of files; and
storing the set of files on the computer.

4. (Once Amended) A method for identifying a set of files and application functionality needed to run the set of files when the computer is disconnected from a network, the method comprising the steps of:
determining the set of files to be stored locally on the computer;
identifying a type for each file in the set of files;
storing the set of files locally on the computer;
for each file, associating the type with a handler routine;
sending each file to the associated handler routine to identify application functionality needed to run each file; and
installing the identified application functionality locally on the computer.

5. (Unchanged) The method of Claim 4, wherein the step of determining the set of files to be stored locally on the computer comprises receiving user input, wherein the user input corresponds to a plurality of files that are to be stored locally on the computer.

6. (Unchanged) The method of Claim 4, wherein the step of determining the set of files to be stored locally on the computer comprises the steps of:
searching a plurality of files in a plurality of storage locations on the computer;

determining whether each file found in the plurality of storage locations is to be stored locally on the computer; and
if so, then adding the file to the set of files.

7. (Cancelled)

8. (Once Amended) The method recited in Claim 4, wherein the handler routine comprises instructions for scanning the associated file and determining the application functionality that is needed to execute the associated file.

9. (Unchanged) The method recited in Claim 8 wherein application functionality comprises programs, features and components.

10. (Unchanged) A method for identifying a set of application functionality to be stored on a computer connected to a network, comprising the steps of:

causing a document identification engine (DIE) to create a list of a plurality of files stored locally on the computer;

sending the list of files from the DIE to a document mapping engine (DME);

causing the DME to identify a proper handler routine for each file in the list of files;

sending each file from the DME to the proper handler routine;

causing the handler routine to identify the application functionality needed to execute each file;

sending a list of needed application program functionality of the handler routine to the DME;

sending a list of needed application program functionality from the DME to a migration engine (ME);

causing the ME to determine the current status of the needed application functionality; and

if the status of the needed application functionality indicates that the needed

application functionality is not installed locally on the computer, then causing the ME to install the needed application functionality to the computer.

11. (Unchanged) A computer-readable medium comprising computer-readable instructions, which when executed, performs the steps of Claim 10.

12. (New) The method of claim 6 wherein the step of determining whether each file found in the plurality of storage locations is to be stored locally is based on a set of rules.

13. (New) The method of claim 6 wherein the step of determining whether each file found in the plurality of storage locations is to be stored locally is based on a user's usage patterns.

14. (New) The method of claim 4 wherein the step of identifying application functionality needed to run each file comprises determining whether each file needs multiple application functionality.

15. (New) The method of claim 14 wherein the step of determining whether each file needs multiple application functionality comprises mapping application functionality to a file embedded in a file in the set of files.

16. (New) The method of claim 15 wherein the embedded file is an Object Linking and Embedding (OLE) object.

17. (New) The method of claim 15 wherein the embedded file is a hyperlink.

18. (New) The method of claim 10 further comprising the steps of:
causing the handler to notify the DME of an embedded file; and
in response to receiving the notification of the embedded file, causing the DME to transmit the embedded file to another handler associated with the embedded file.